Mid-Semester Check In 2

2025-07-19

Load Libraries

```
#Always good to get all the packages up and running before doing anything else
library(data.table)
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.4     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::between()     masks data.table::between()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ dplyr::first()       masks data.table::first()
## ✖ lubridate::hour()    masks data.table::hour()
## ✖ lubridate::isoweek() masks data.table::isoweek()
## ✖ dplyr::lag()         masks stats::lag()
## ✖ dplyr::last()        masks data.table::last()
## ✖ lubridate::mday()    masks data.table::mday()
## ✖ lubridate::minute()  masks data.table::minute()
## ✖ lubridate::month()   masks data.table::month()
## ✖ lubridate::quarter() masks data.table::quarter()
## ✖ lubridate::second()  masks data.table::second()
## ✖ purrr::transpose()   masks data.table::transpose()
## ✖ lubridate::wday()    masks data.table::wday()
## ✖ lubridate::week()    masks data.table::week()
## ✖ lubridate::yday()    masks data.table::yday()
## ✖ lubridate::year()    masks data.table::year()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

Load metadata and gene expression data and

created a combined data frame.

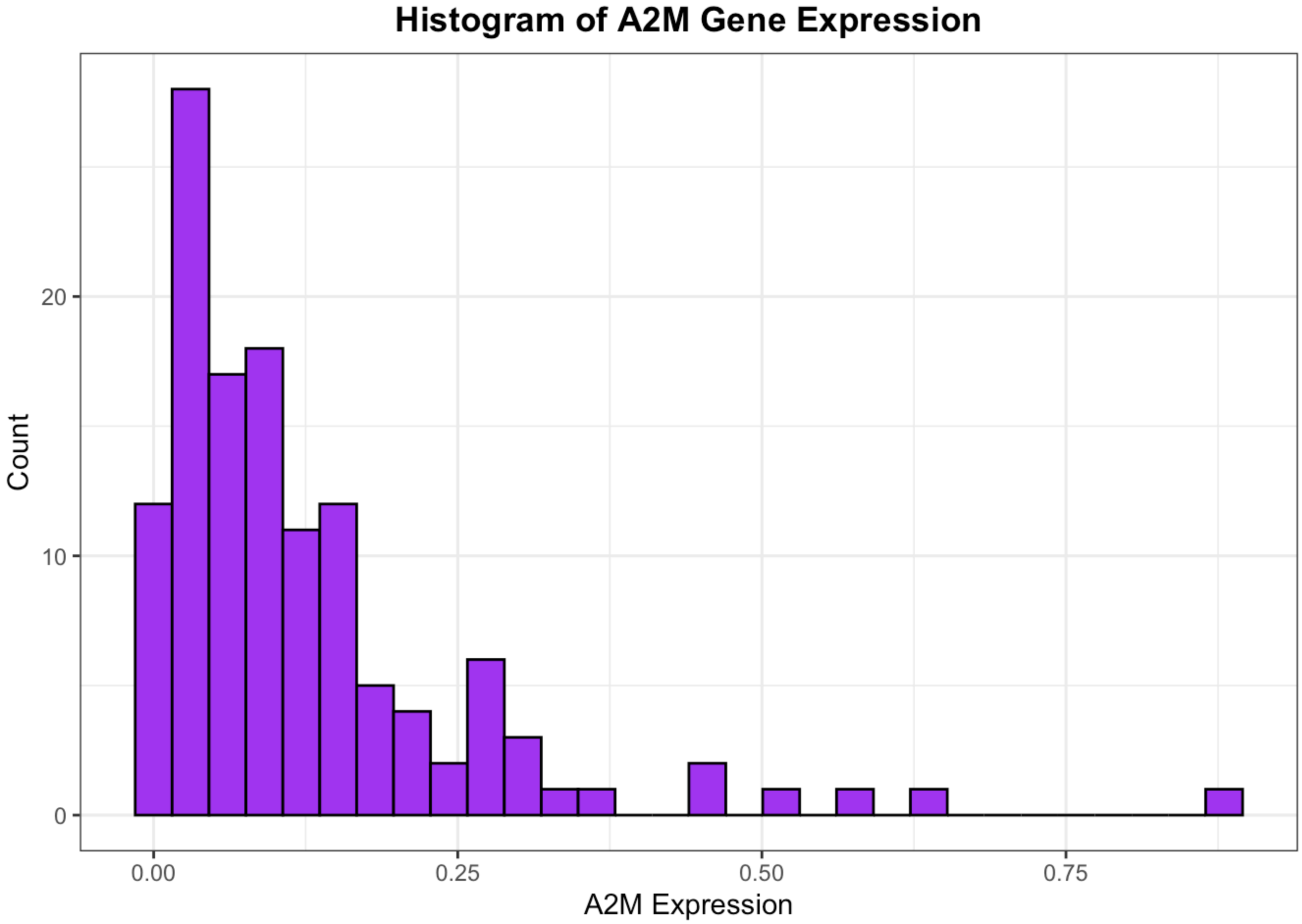
```
filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_series_matrix-1.csv"
meta_data <- fread(filename)

filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_genes.csv"

#Makes so that V1 col that contains the gene names are now the row names
gene_expr <- fread(filename) %>%
  column_to_rownames("V1") %>%
 #Transposing with make the table be 126 observations by 100 observations
  t() %>%
  data.frame()

#Create combined data frame. Now we have one single data frame with all the data. I didn't initially know how to do this, so I googled "how to combine objects into rows and columns in R" and used the following website for reference: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/cbind
comb_df <- cbind(meta_data, gene_expr)
```

#Started making plots

```
#Created a function with four parameters. The first one was df, second genes, etc. The df was for data frame, second was for the actual gene list, continuous_cov was for the numeric covariate Charlson score, and cat_cov was for our categorical covariates sex and disease status.
create_gene_plots <- function(df, genes, continuous_cov, cat_cov){
#I found out that having strings passed into ggplot causes an error. I put this into chat.dartmouth.edu and it told me that "!!sym()" can be used to refer to a column when its name is stored as a string and that "!!" unquotes the symbol. So I used that as a reference for the below codes.
#I used "lapply" to create a loop that let me create plots for each gene.
  hist_plots <- lapply(genes, function(gene){
    ggplot(data = df, mapping = aes(x = !!sym(gene))) +
    geom_histogram(fill = "purple", col = "black") +
    #Paste helps combine strings so I used that here and for the plots below.
    labs(x = paste(gene, "Expression"), y = "Count", title = paste("Histogram of", gene, "Gene Expression")) +
    theme_bw() +
    theme(
       plot.title = element_text(face = "bold", hjust = 0.50)
    )
  })

  #Scatterplots
  #"str_to_title" used to capitalize the first letter of each word. I did not know this, so I googled how to do this and used the following website for guidance: https://stringr.tidyverse.org/reference/case.html. I knew that "gsub" could be used to substitute spaces for underscores. I did these simply to make the plots look cleaner.
  cont_cov_label <- str_to_title(gsub("_", " ", continuous_cov))
  #Similar to the histogram, used "lapply" to create a loop that let me create plots for each gene. From here onwards, I used what I did above with the histogram to finish off the rest of the plot, only changing axes, etc. as necessary.
  scatter_plots <- lapply(genes, function(gene){
    ggplot(data = df, mapping = aes(x = !!sym(gene), y = !!sym(continuous_cov))) +
    geom_point(color = "blue", size = 2) +
    labs(
     x = paste(gene, "Expression"), y = cont_cov_label,
     title = paste("Scatterplot of", cont_cov_label, "vs", gene, "Expression")
    ) +
    theme_bw() +
    theme(
     plot.title = element_text(face = "bold", hjust = 0.50)
    )
  })



  #Boxplots
  #Again used "str_to_title" and "gsub" to clean up the title names for sex and disease status.
  cat_cov_label1 <- str_to_title(gsub("_", " ", cat_cov[1]))
  cat_cov_label2 <- str_to_title(gsub("_", " ", cat_cov[2]))
  box_plots <- lapply(genes, function(gene){
    ggplot(data = df, mapping = aes(x = !!sym(cat_cov[1]), y = !!sym(gene), fill = !!sym
(cat_cov[1]))) +
  geom_boxplot() +
  #Had to replace the ~ with the vars function for the faceting to work, otherwise I k
ept getting an error. This was just trial and error until it worked.
   facet_wrap(vars(!!sym(cat_cov[2]))) +
  #Again just used what I did above and changed labels as necessary.
   labs(
   x = cat_cov_label1, y = paste(gene, "Expression"),
   title = paste("Boxplot of", gene, "Expression by", cat_cov_label1, "and", cat_cov_
label2),
   fill = cat_cov_label1
   ) +
  theme_bw() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
 })


 #Created a list of "things" to return.
 return(list(
  "Histograms" = hist_plots, "Scatterplots" = scatter_plots, "Boxplots" = box_plots
 ))
}

#Doing it this way will update the title, axis, and gene that was plotted. Assigned the
output of our function to a variable named res1. You should see res1 in our environment.
If you print res1, you'll get all the plots.
res1 <- create_gene_plots(
 df = comb_df, genes = c("A2M", "AASDHPPT", "AAAS"),
 continuous_cov = "charlson_score",
 cat_cov = c("sex", "disease_status")
)

#Printing histograms, scatterplots, and boxplots separately.
res1[["Histograms"]]
```

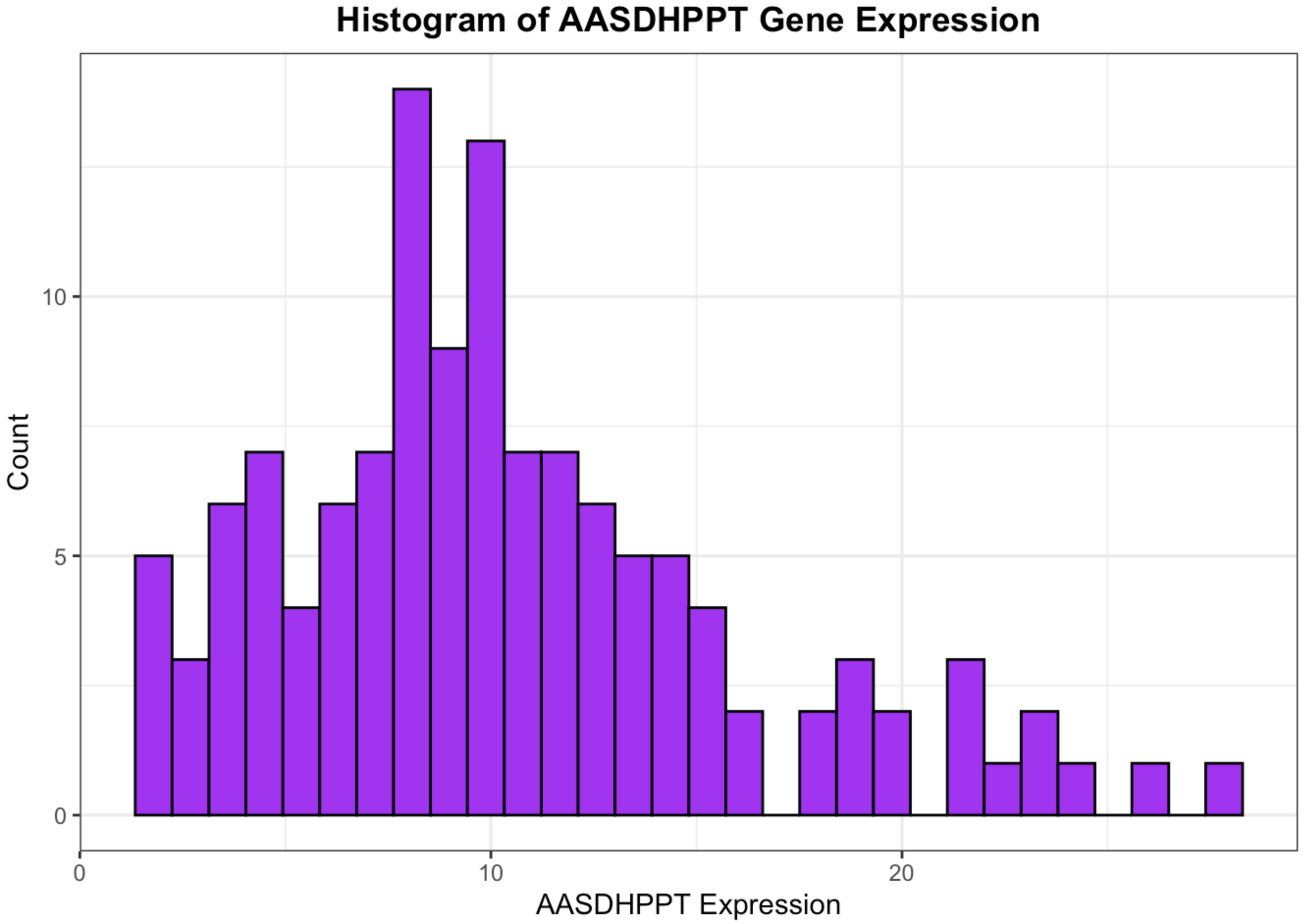
```
## [[1]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
##
## [[2]]
```

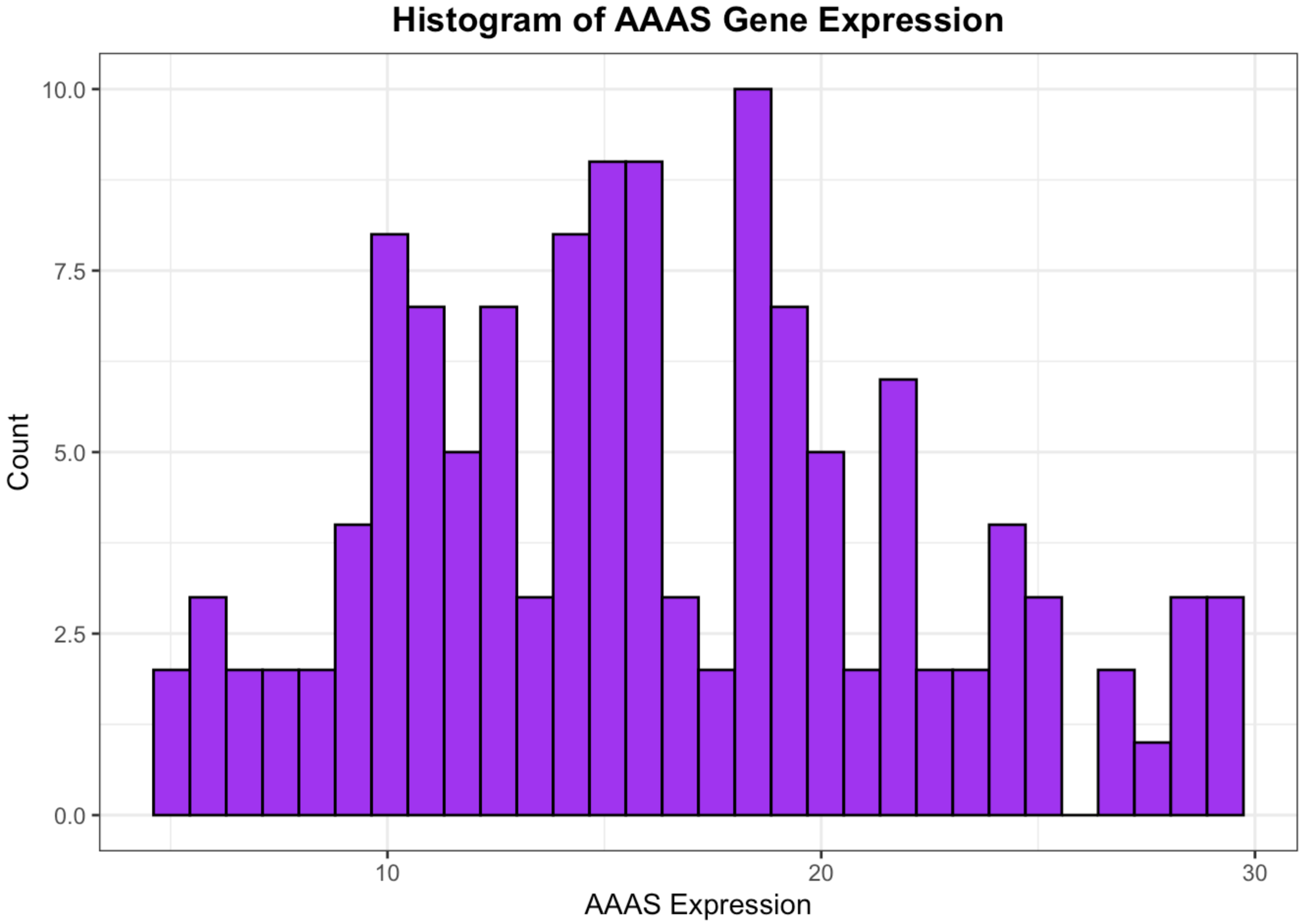
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram of AASDHPPT Gene Expression

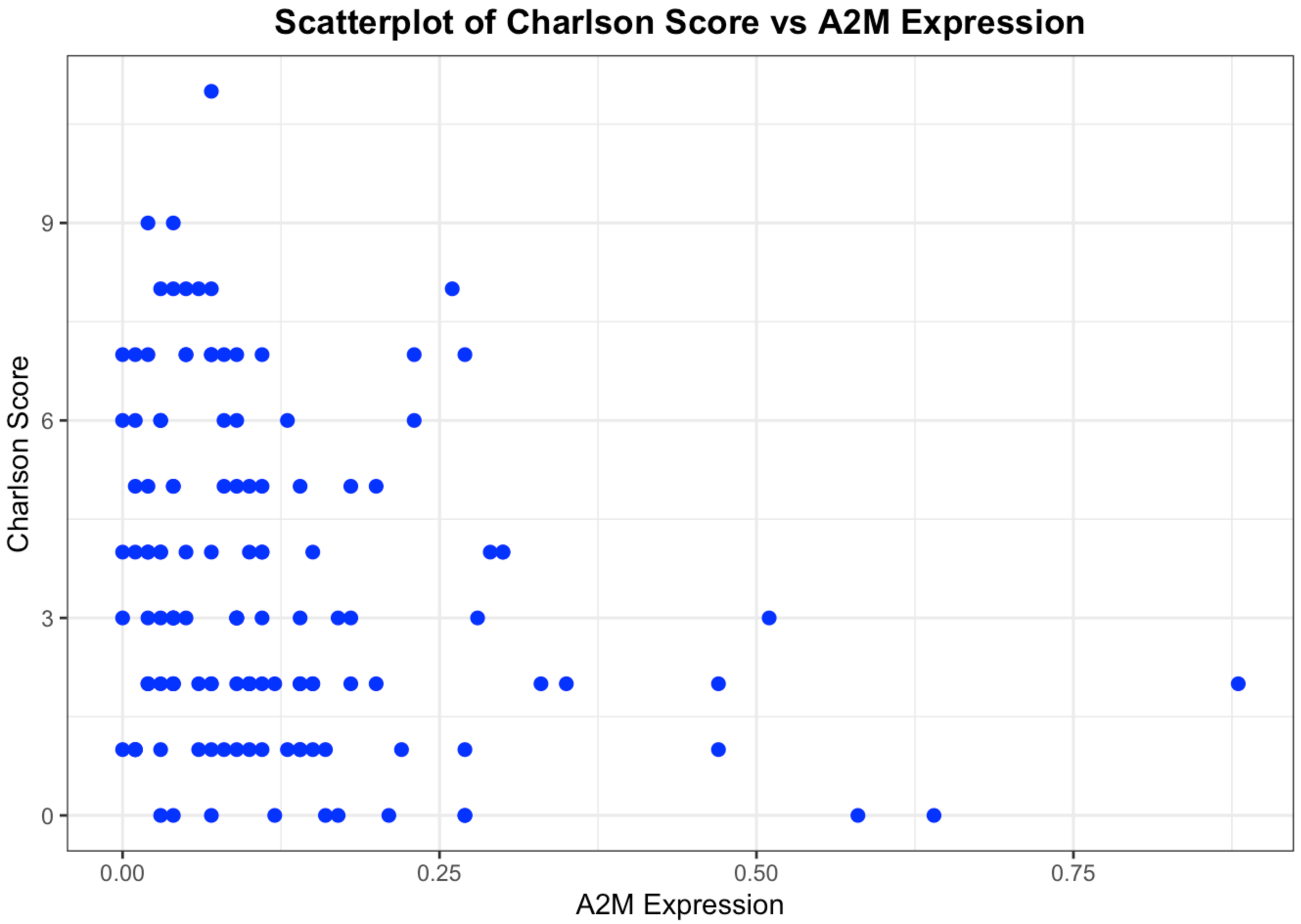
```
##
## [[3]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

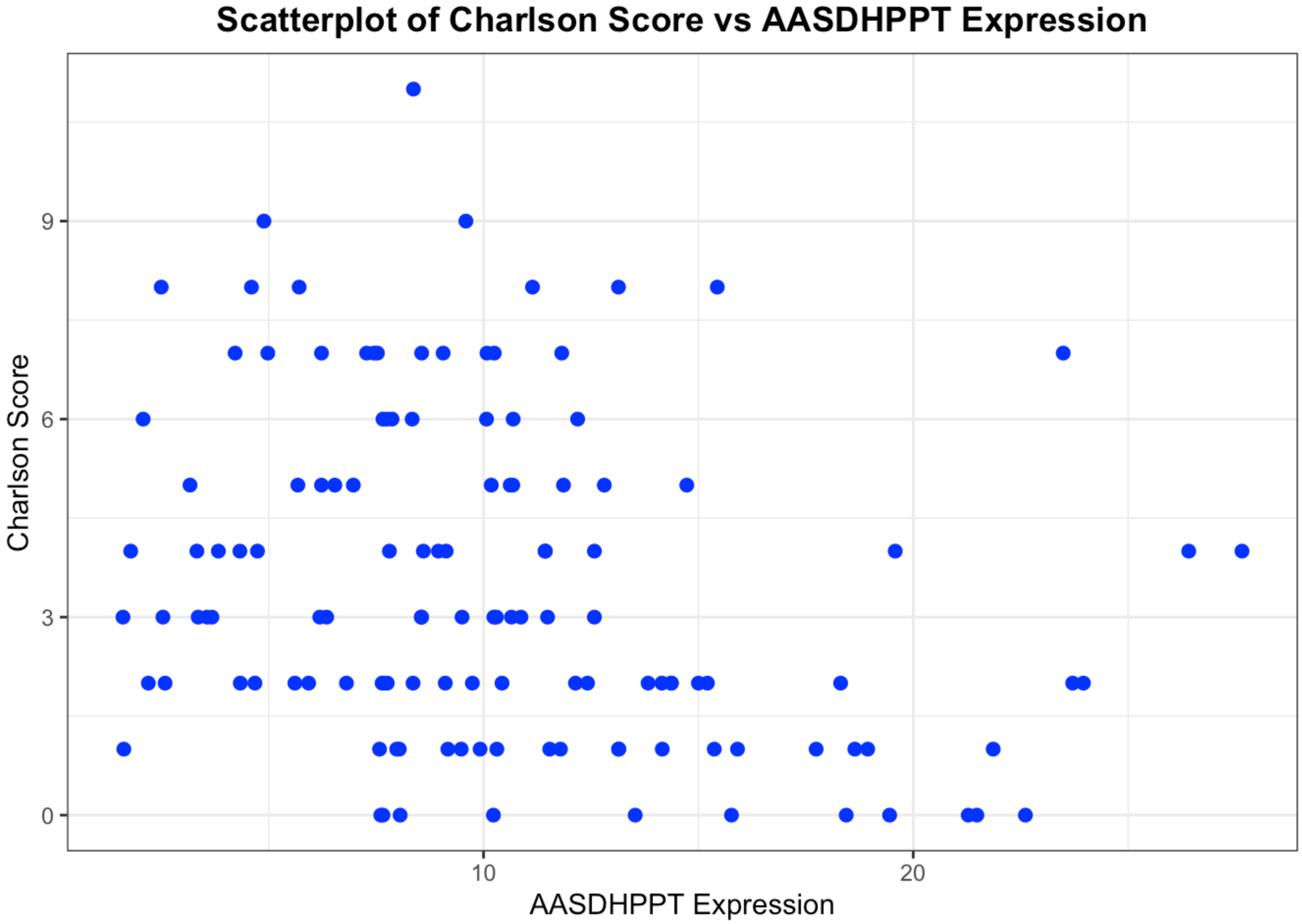


```
res1[["Scatterplots"]]
```

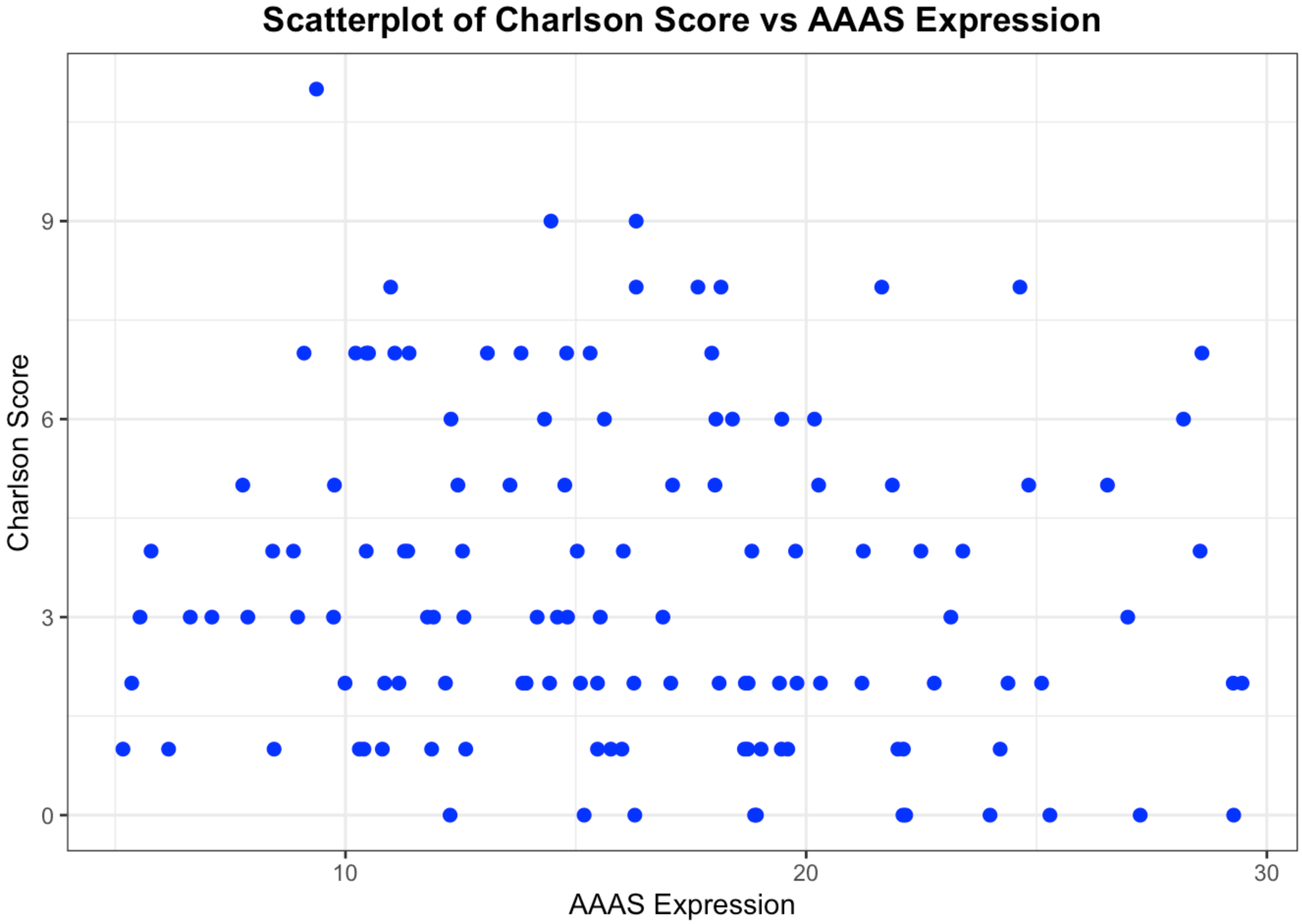
```
## [[1]]
```


Scatterplot of Charlson Score vs A2M Expression
Charlson Score
9
6
3
0
0.00
0.25
0.50
0.75
A2M Expression

```
##
## [[2]]
```

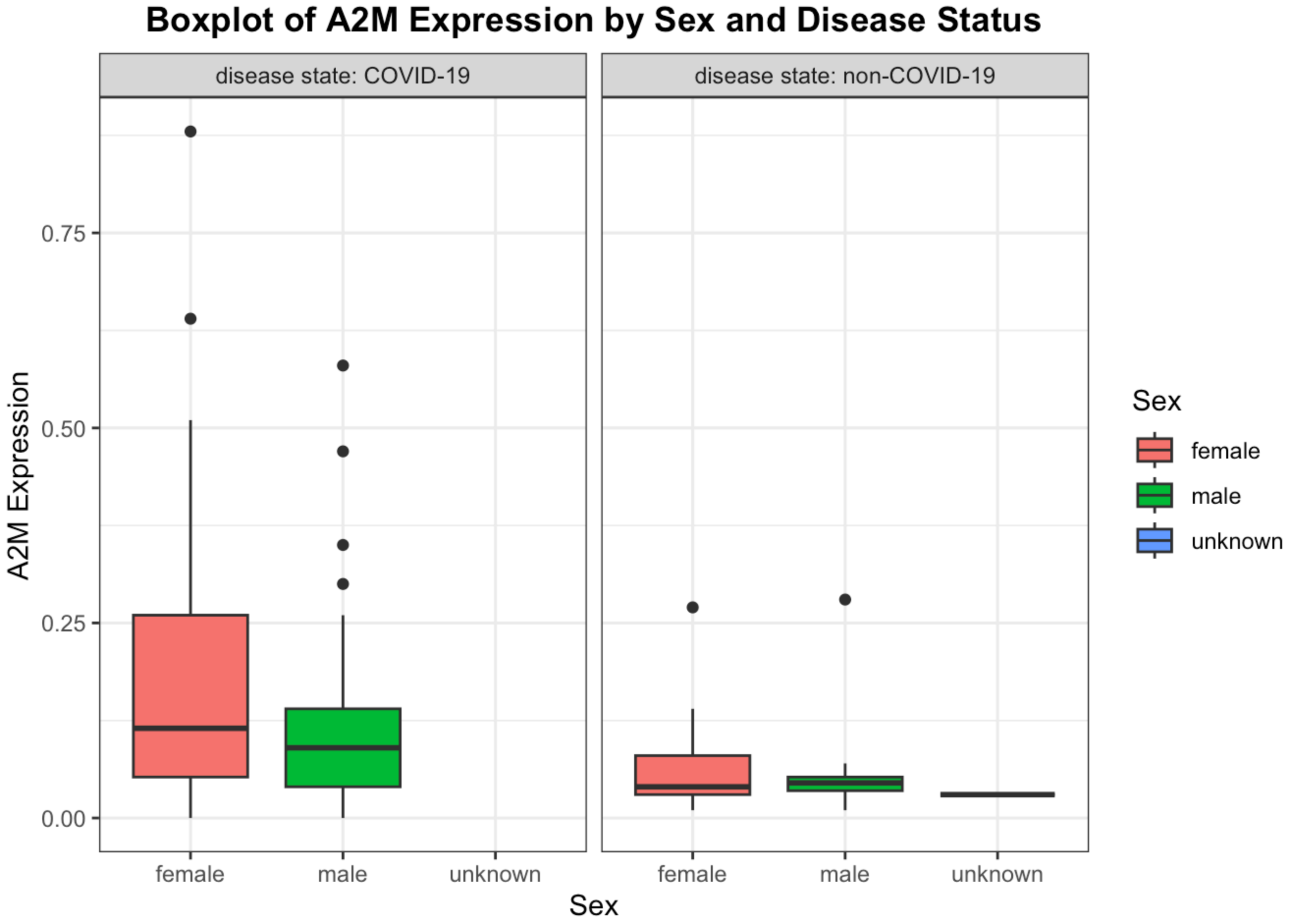

Scatterplot of Charlson Score vs AASDHPPT Expression
Charlson Score
0
3
6
9
10
20
AASDHPPT Expression

```
##
## [[3]]
```

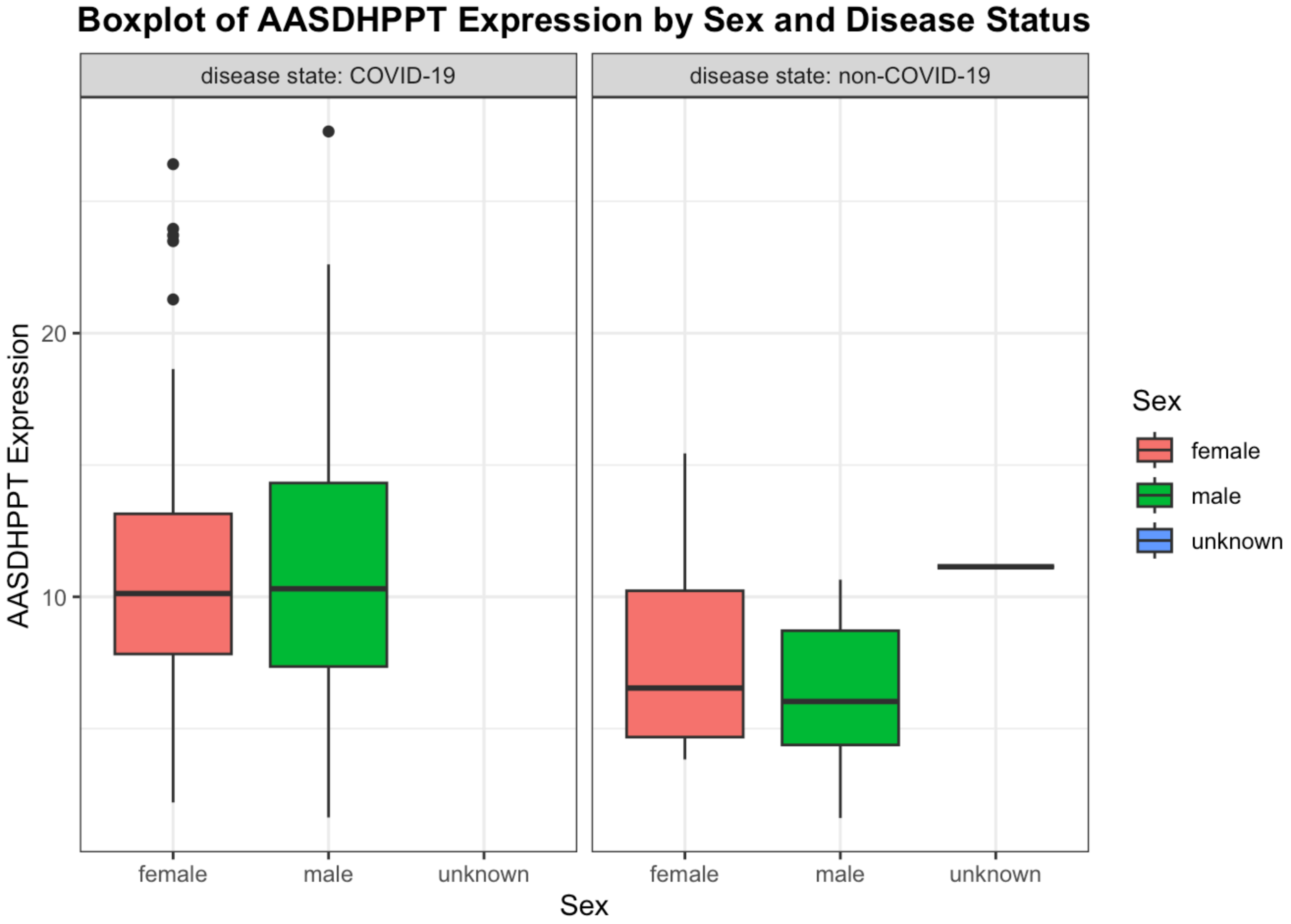


```
res1[["Boxplots"]]
```

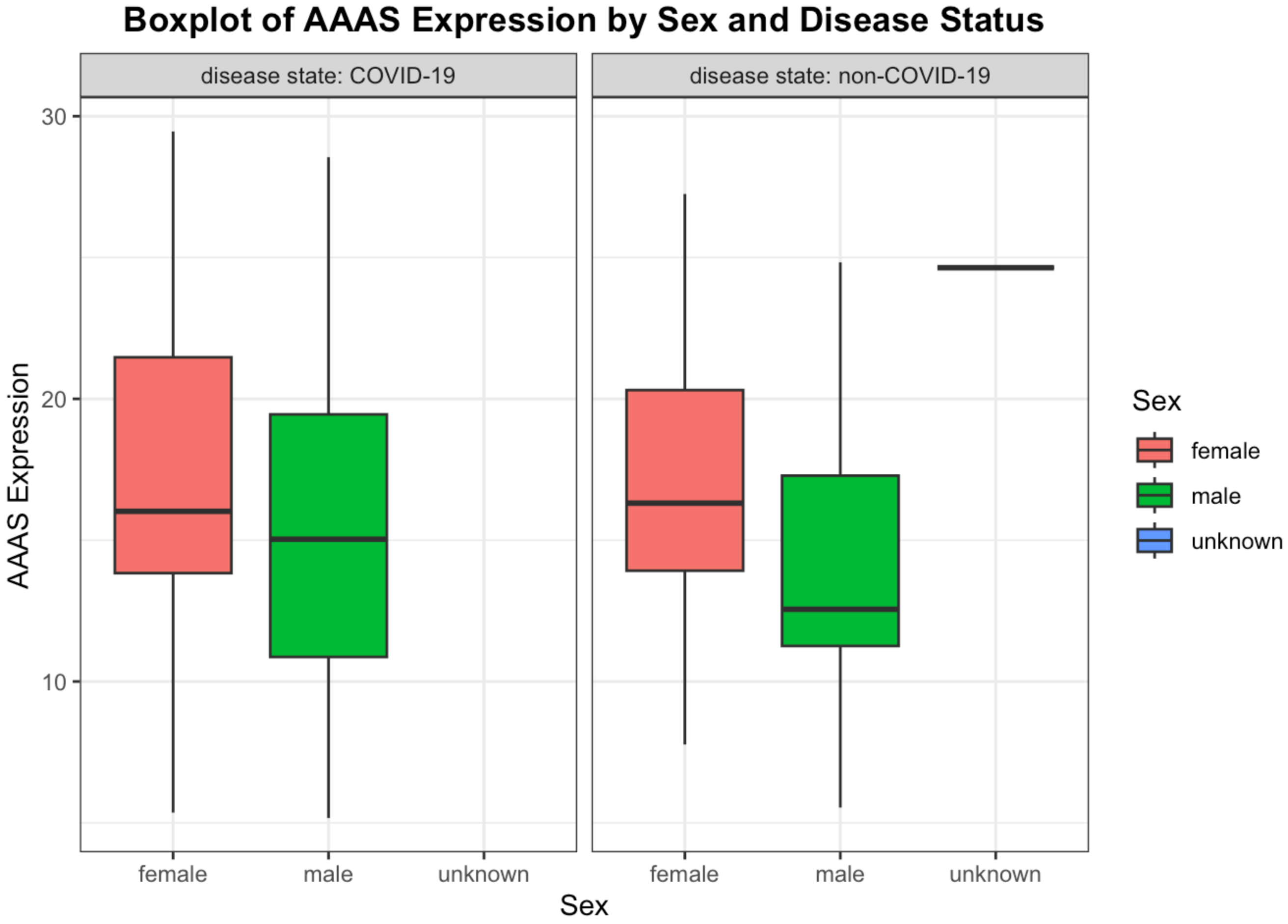
```
## [[1]]
```


Boxplot of A2M Expression by Sex and Disease Status
disease state: COVID-19
disease state: non-COVID-19
A2M Expression
0.00
0.25
0.50
0.75
female
male
unknown
female
male
unknown
Sex
Sex
female
male
unknown

```
##
## [[2]]
```


Boxplot of AASDHPPT Expression by Sex and Disease Status
disease state: COVID-19
disease state: non-COVID-19
AASDHPPT Expression
20
10
female
male
unknown
female
male
unknown
Sex
Sex
female
male
unknown

```
##
## [[3]]
```



#Note: All references used in this assignment are attached in the comment above the codes for which I used outside sources. I also used the lecture notes to help with this assignment.